Modeling of Lithium-ion Batteries via Tensor-Network-Based Volterra Model

Yangsheng Hu * **Raymond A. de Callafon** * **Ning Tian** ** **Huazhen Fang** **

* *Department of Mechanical and Aerospace Engineering, University of California, San Diego, CA 92094 USA (e-mail: {yah071, callafon}@ucsd.edu).*

** *Department of Mechanical Engineering, University of Kansas, Lawrence, KS 66045 USA (e-mail: {ning.tian, fang}@ku.edu)*

Abstract: Equivalent circuit models (ECMs) are a popular and important tool to characterize and safely use lithium-ion batteries, due to their parsimonious structure, fast computation, and physical interpretability. However, they are often limited in predictive accuracy and thus insufficient for some critical applications. To overcome the limitation, this paper proposes to integrate the linear double-capacitor model, an ECM, with a data-based Volterra model to build a physics-informed data-driven model for lithium-ion batteries, which is named as *Volterra double-capacitor* (VDC) model. The VDC model uses the ECM as a feature extractor to capture physical features of charging/discharging; taking the features, the Volterra model then approximates the complex nonlinear dynamics inherent to the battery and predicts the terminal voltage. In particular, the Volterra model exploits a tensor network representation to break the curse of dimensionality. Further, a parameter identification approach is constructed to extract the parameters of the model from data. The experimental validation demonstrates this new model's high accuracy, suggesting its promise for various future applications.

Keywords: Battery modeling, equivalent circuit model, Volterra model, tensor network, system identification, machine learning.

1. INTRODUCTION

Lithium-ion batteries (LiBs) have found wide use in numerous applications ranging from portable electronics to electric vehicles and grid-scale energy storage. The pursuit for better LiB technologies has gained ever-increasing momentum in both academia and industry (Goodenough and Park, 2013; Plett, 2015; Scrosati et al., 2015; Rahn and Wang, 2013; Liu et al., 2019). The development of advanced battery management systems (BMSs) is at the forefront of related research, due to their essential role in ensuring the operating performance and safety of LiBs Wang et al. (2017).

Mathematical models that describe LiB dynamics are fundamental for BMS algorithms. The literature has intensively investigated two main types of battery models: electrochemical models, and equivalent circuit models (ECMs) (Santhanagopalan et al., 2006; Di Domenico et al., 2010; Forman et al., 2012; Pinson and Bazant, 2012; Hatzell et al., 2012; Moura et al., 2017; Lin et al., 2014; Xia et al., 2016; Farmann and Sauer, 2016; Rahn and Wang, 2013; Ramadesigan et al., 2012; Li et al., 2018; Zou et al., 2018; Plett, 2015; Tian et al., 2021). The electrochemical models build upon electrochemical principles to characterize the electrochemical reactions, Li-ion diffusion, and concentration and potential changes in the electrode/electrolyte during charging/discharging of LiBs. They usually offer high fidelity and descriptive accuracy but also require expensive computation. ECMs present electrical-circuit-based analogs to reproduce a LiB's current/voltage dynamics. They have simple structures, which, in turn, lead to considerable computational efficiency. The circuit-based structures also allow interpretation from a physical perspective, so one can easily associate them with the characteristics and phenomenological observations of LiBs. These merits make ECMs a favorable, and increasingly popular, choice in practical battery engineering. Yet, a challenge for ECMs is their relatively limited accuracy due to the simple structures, which restricts their application to a certain extent.

Both electrochemical models and ECMs can be considered as physics-based models. Despite many recent research advances, they are still found inadequate to capture the full spectrum of LiBs' complex dynamics resulting from a mix of electrochemical, thermal, and electrical processes and parasite reactions, especially when there are constraints on computational power. Given that today's LiB systems generate much data in operation, pure data-driven approaches from the perspective of system identification or machine learning provide a valuable alternative to LiB modeling (Ljung, 1999; Verhaegen and Verdult, 2007; Murphy, 2012; Hu and de Callafon, 2017; Hu et al., 2020). Skipping the underlying physics and peculiarities, these approaches extract black-box models to relate the input and output data of LiBs by referring to statistical and optimization methods. A few relevant investigations can be found in

Murphey et al. (2012); Hu et al. (2015); Yang et al. (2018); Chemali et al. (2018); Hu et al. (2018); Ardeshiri et al. (2020); Hannan et al. (2020). With a data-driven nature, these models can be conveniently constructed from data to show good fitting accuracy. Nonetheless, their performance can be limited by the data richness and informativeness and may also suffer from the overfitting issue in training.

This study identifies the complementarity between physics- and data-based modeling and proposes to harness their combined strengths to attain LiB modeling with both high predictive accuracy and structural efficiency. To this end, we integrate an ECM with a data-based Volterra model. The obtained hybrid model by structure consists of a linear and a nonlinear part in cascade. The linear part is a linear double-capacitor model, which is an ECM. It serves as a feature extractor to produce physics-based features characteristic of the charging/discharging. The nonlinear part is made up of a Volterra model based on the tensor network (TN) representation to simulate the nonlinear dynamic relationship between the produced features and the terminal voltage. Note that the TN representation will allow one to exploit a Volterra model of high degree to enhance the approximation of nonlinearity, while breaking the curse of dimensionality through trading storage for computation. In addition, the estimation of the linear and nonlinear part is presented in this paper. A low-rank TN representation of the Volterra model is obtained. The proposed hybrid model, named as *Volterra Double-Capacitor* (VDC) model, illustrates high accuracy in terminal voltage prediction. The design of the VDC model is partly inspired by the nonlinear double-capacitor (NDC) model (Tian et al., 2021). The NDC model couples a linear double-capacitor circuit with a nonlinear voltage source to simulate the charge diffusion inside an electrode and the nonlinear voltage behavior simultaneously. Compared to the NDC model, the proposed VDC study takes a data-driven viewpoint and leverages the powerful Volterra model to describe the complicated nonlinearity ingrained in LiBs, while still maintaining a relatively simple structure.

The rest of this paper is organized as follows. Section 2 introduces the basic NDC model and the TN-based Volterra model. The VDC model is formulated in Section 3. The corresponding parameter estimation for both the linear and nonlinear parts of the VDC model is presented in Section 4. Section 5 illustrates the experimental validation results. Finally, Section 6 summarizes this paper.

2. OVERVIEW OF THE NDC MODEL AND TN-BASED VOLTERRA MODEL

This section will begin with an overview of the NDC model and then proceed to describe the TN-based Volterra model. The introduction will lay a foundation for the development of the VDC model in Section 3.

2.1 The Basic Nonlinear Double-Capacitor Model

The NDC model was first proposed in Tian et al. (2021), a basic version of which is shown in Fig. 1. The model is a cascade of a pair of parallel RC circuits and a nonlinear voltage source. The RC circuits simulate an electrode of a LiB cell. Specifically, R_b-C_b represents an analogy to the bulk inner part of the electrode, and R_s-C_s corresponds to the surface area; the charge transfer between C_b and C_s mimics the diffusion of lithium ions within the electrode. The nonlinear voltage source $U = h(V_s)$ depicts the battery's nonlinear voltage behavior, where $h(\cdot)$ is a static nonlinear function. The terminal voltage is the sum of U and the voltage across the internal resistance R_0. The NDC model generalizes a linear double-capacitor model proposed in Johnson et al. (2001) to take into account a LiB cell's inherent nonlinearity. It can also be justified as an approximation of the single particle model, which is a reduced-order electrochemical model.

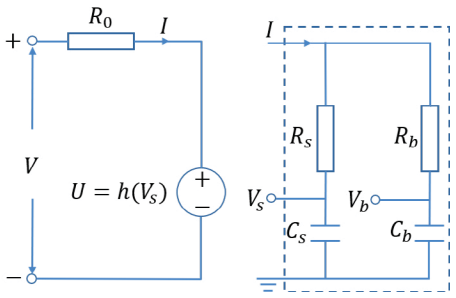


Fig. 1. The basic NDC model.

For the NDC model, the dynamics of the parallel RC circuit can be expressed as

$$\begin{bmatrix} \dot{V}_b(t) \\ \dot{V}_s(t) \end{bmatrix} = A \begin{bmatrix} V_b(t) \\ V_s(t) \end{bmatrix} + BI(t) \tag{1}$$

where I is the applied current ($I > 0$ for charging, and $I < 0$ for discharging), V_b and V_s are the voltages across C_b and C_s, respectively, and

$$A = \begin{bmatrix} \frac{-1}{C_b(R_b+R_s)} & \frac{1}{C_b(R_b+R_s)} \\ \frac{1}{C_s(R_b+R_s)} & \frac{-1}{C_s(R_b+R_s)} \end{bmatrix}, B = \begin{bmatrix} \frac{R_s}{C_b(R_b+R_s)} \\ \frac{R_b}{C_s(R_b+R_s)} \end{bmatrix}$$

Here, $V_b = V_s = 1$ V when the cell is at full charge (SoC = 1); $V_b = V_s = 0$ V at full depletion (SoC = 0). Here, SoC is specifically defined as

$$\text{SoC} = \frac{C_b V_b + C_s V_s}{C_b + C_s} \tag{2}$$

Based on (1), $V_s(t)$ is governed by the following discrete-time transfer function in a zero-order-hold setting:

$$V_s(t) = \Big(\underbrace{\frac{\beta_1 q^{-1}}{1-q^{-1}}}_{G_1(q^{-1})} + \underbrace{\frac{\beta_2 q^{-1}}{1-\beta_3 q^{-1}}}_{G_2(q^{-1})}\Big) I(t) + V_s(0) \tag{3}$$

where q^{-1} is the backward time-shift operator, $G_1(q^{-1})$ is an integrator, and $G_2(q^{-1})$ is a first-order filter. Note that $t = k\Delta T$ for $k \in \mathbb{Z}^+$, where ΔT is the sampling interval. The terminal voltage is given by

$$V(t) = h(V_s(t)) + I(t)R_0$$

2.2 TN-Based Representation of a Volterra Model

A discrete-time Volterra model of degree d with the input $\mathbf{u}(t) \in \mathbb{R}^p$ and the output $y(t) \in \mathbb{R}$ can be expressed as

$$y(t) = c_0 + \sum_{i=1}^{d} \sum_{k_1,\cdots,k_i=0}^{M-1} \sum_{\alpha_1,\cdots,\alpha_i=0}^{p-1} c_{1:i}^{k,\alpha} \phi_{1:i}^{k,\alpha}(\mathbf{u}(t)) \tag{4}$$

where $\phi_{1:i}^{k,\alpha}(\cdot) \in \mathbb{R}$ is the ith kernel function, $c_{1:i}^{k,\alpha} \in \mathbb{R}$ the kernel coefficient, and M the memory length. Further,

$$\phi_{1:i}^{k,\alpha}\left(\mathbf{u}\left(t\right)\right) = \prod_{j=1}^{i} u^{(\alpha_j+1)}\left(t-k_j\right) \tag{5}$$

$$c_{1:i}^{k,\alpha} = \mathbf{c}_i\left(k_1,\alpha_1;k_2,\alpha_2;\cdots;k_i,\alpha_i\right) \tag{6}$$

Here, $u^{(\alpha_j+1)}$ is the (α_j+1)th element of $\mathbf{u}$.

Capable of acting on a high-dimensional space, the Volterra model can provide strong representation power to characterize nonlinear relationship in data, with increasing application across various disciplines (Wambacq and Sansen, 1998; Azpicueta-Ruiz et al., 2011; Stoudenmire and Schwab, 2016; Burt and de Morais Goulart, 2018; Hu et al., 2019). Here, it will be exploited as a data-driven nonlinear mapping, playing a role that is played by $h(\cdot)$ in the NDC model, to capture the complicated, elusive nonlinearities ingrained in LiBs. However, the number of parameters in (4) will increase exponentially when d goes up, making the storage requirement grow exponentially and causing the curse of dimensionality (Batselier et al., 2017). This limits the application of the Volterra model to efficiently grasp highly nonlinear phenomena.

A TN representation of a Volterra model will resolve the issue. A d-way tensor is denoted as $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{n_1\times n_2\times\cdots\times n_d}$. Each entry of $\boldsymbol{\mathcal{T}}$ is denoted as $\boldsymbol{\mathcal{T}}^{(i_1 i_2\cdots i_d)}$ via d integer indices $(i_1 i_2\cdots i_d)$.

Definition 1. (The k-Mode Product (Oseledets, 2011)) For a tensor $\boldsymbol{\mathcal{T}} \in \mathbb{R}^{n_1\times\cdots\times n_k\times\cdots\times n_d}$ and a matrix $\mathbf{U} \in \mathbb{R}^{p_k\times n_k}$, the k-mode product $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{T}} \times_k \mathbf{U}$ is defined as

$$\boldsymbol{\mathcal{X}}^{(i_1\cdots i_{k-1} j i_{k+1}\cdots i_d)} = \sum_{i_k=1}^{n_k} \mathbf{U}^{(j i_k)} \boldsymbol{\mathcal{T}}^{(i_1\cdots i_{k-1} i_k i_{k+1}\cdots i_d)}$$

and $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{n_1\times\cdots\times n_{k-1}\times p_k\times n_{k+1}\times\cdots\times n_d}$.

Definition 2. (Tensor Train (TT) Decomposition (Oseledets, 2011)) The d-way tensor $\boldsymbol{\mathcal{T}}$ can be represented by a linear TN such that

$$\boldsymbol{\mathcal{T}}^{(i_1 i_2\cdots i_d)} = \sum_{\alpha_0,\cdots,\alpha_d} \boldsymbol{\mathcal{T}}_1^{(\alpha_0 i_1 \alpha_1)} \boldsymbol{\mathcal{T}}_2^{(\alpha_1 i_2 \alpha_2)} \cdots \boldsymbol{\mathcal{T}}_d^{(\alpha_{d-1} i_d \alpha_d)}$$

where $\boldsymbol{\mathcal{T}}_1,\cdots,\boldsymbol{\mathcal{T}}_d$ are called TT-cores. Each $\boldsymbol{\mathcal{T}}_k$ is a 3-way tensor of dimensions $r_{k-1}\times n_k\times r_k$, where r_{k-1}, r_k are called the TT-ranks and $r_0 = r_d = 1$.

As in Batselier et al. (2017), one can incorporate all the kernel coefficients in (6) into a $(d+1)$-way Volterra tensor $\boldsymbol{\mathcal{V}} \in \mathbb{R}^{(pM+1)\times\cdots\times(pM+1)\times 1}$ such that

$$\begin{aligned} y\left(t\right) &= \boldsymbol{\mathcal{V}} \times_1 \mathbf{u}_t^{\mathrm{T}} \times_2 \mathbf{u}_t^{\mathrm{T}} \cdots \times_d \mathbf{u}_t^{\mathrm{T}} \\ &= \left(\boldsymbol{\mathcal{V}}_1 \times_2 \mathbf{u}_t^{\mathrm{T}}\right)\left(\boldsymbol{\mathcal{V}}_2 \times_2 \mathbf{u}_t^{\mathrm{T}}\right)\cdots\left(\boldsymbol{\mathcal{V}}_d \times_2 \mathbf{u}_t^{\mathrm{T}}\right) \end{aligned} \tag{7}$$

where

$\mathbf{u}_t = \left[1, \mathbf{u}^{\mathrm{T}}\left(t\right), \mathbf{u}^{\mathrm{T}}\left(t-1\right), \cdots, \mathbf{u}^{\mathrm{T}}\left(t-M+1\right)\right]^{\mathrm{T}} \in \mathbb{R}^{pM+1}$ and $\{\boldsymbol{\mathcal{V}}_1,\cdots,\boldsymbol{\mathcal{V}}_d\}$ are the TT-cores of the Volterra tensor $\boldsymbol{\mathcal{V}}$. The TT-cores $\boldsymbol{\mathcal{V}}_i \in \mathbb{R}^{r_{i-1}\times(pM+1)\times r_i}$ with $r_0 = r_d = 1$ and $\{r_1,\cdots,r_{d-1}\}$ to be identified.

With the above formulation, rather than estimating all the parameters in (6) at prohibitive computational costs due to the curse of dimensionality, one just needs to estimate its dual TN representation $\{\boldsymbol{\mathcal{V}}_1,\cdots,\boldsymbol{\mathcal{V}}_d\}$ whose storage requirement is $O((d-1)r^2(pM+1)+r(pM+1))$ with $r = \max\{r_1,\cdots,r_{d-1}\}$. The storage requirement can be greatly reduced if r is sufficiently small. The computational complexity of using (7) is approximately $O(d(pM+1)r^2)$.

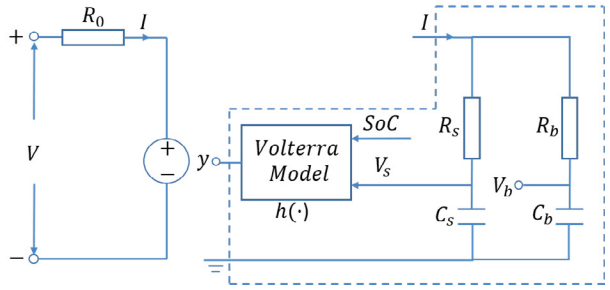


Fig. 2. The structure of the proposed VDC model.

3. FORMULATION OF THE VDC MODEL

This section focuses on formulating the VDC model by drawing upon the NDC model and TN-based Volterra model. The motivation lies in the key insight that, even though the NDC model demonstrates effectiveness for LiB modeling, there is still a space for improvements. First, the NDC model considers only a simple static nonlinear function $h(\cdot)$ and neglects other dynamic nonlinear phenomena, e.g., hysteresis. Hence, it uses $h(\cdot)$ to implicitly enforce a trade-off in describing both static and dynamic nonlinear behaviors of a cell, at the sacrifice of actual predictive accuracy. For example, a simple choice for a static $h(\cdot)$ such as the SoC-OCV mapping may be appropriate when the battery is in an idling period. But the prediction will be less accurate during charging or discharging. Second, only V_s-dependent nonlinearity is addressed in the NDC model, where V_s can be viewed as a physical feature of charging/discharging. Yet, a LiB cell's nonlinear dynamics is practically more diverse, implying a need for using a selection of more useful features.

Based on the above, we propose the VDC model, which introduces two crucial modifications to enhance the prediction performance. First, the TN-based Volterra model in (7) is selected as the nonlinear function $h(\cdot)$. Second, $\mathrm{SoC}(t)$ is included as another input feature besides $V_s(t)$. The structure of the VDC model is shown in Fig. 2. Denoting the output of the Volterra model as $y(t) \in \mathbb{R}$, the terminal voltage based on the model is then given by

$$V(t) = y(t) + I(t)R_0 \tag{8}$$

The introduction of SoC as a feature explicitly makes the VDC model to be SoC-dependent. This is reasonable and necessary, since the SoC dependence of the nonlinear behaviors is a well-known attribute of LiBs. Note that $V_s(t)$ will be varying around $\mathrm{SoC}(t)$ during charging/discharging and $V_s(t) = \mathrm{SoC}(t)$ when the battery is fully at rest. It is worth noting that the static and dynamic behaviors are now separated by the interactive behavior of $V_s(t)$ and $\mathrm{SoC}(t)$. The TN-based Volterra model hence can capture not only complicated nonlinear SoC-dependent behaviors, but also the couplings between $\mathrm{SoC}(t)$ and $V_s(t)$.

4. PARAMETER ESTIMATION

In this section, we develop a parameter estimation approach for the proposed VDC model. The approach in-

cludes a two-step procedure. The first step is to estimate the parameters of the linear part, including β_1, β_2, β_3, and R_0. Then, the $V_s(t)$ and SoC(t) can be computed as the input features for $h(\cdot)$. The second step is the estimation of the nonlinear part, where the TT-cores $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d\}$ of the Volterra model are obtained.

4.1 Estimation of the Linear Part

The instantaneous resistance R_0 can be roughly determined using the abrupt voltage jump when there is a step change in the input current. We will use a constant R_0 and the nonlinear correction on R_0 as a function of SoC will be implicitly embedded into $h(\cdot)$.

The estimation of β_1, β_2, β_3 is equivalent to approximate $G_1(q^{-1})$ and $G_2(q^{-1})$ in (3) from the input/output measurement data $\{I(t), V(t)\}$. The basic NDC model structure can be used to initialize the parameter estimation. Select the SoC-OCV curve for the basic NDC model such that OCV $= h_{\text{init}}(\text{SoC})$, where $h_{\text{init}}(\cdot)$ is an initial static nonlinear function for $h(\cdot)$. Then, one can derive the approximation of $V_s(t)$, denoted as $\widehat{V}_s(t)$, from (3)

$$\widehat{V}_s(t) = h_{\text{init}}^{-1}\left(V(t) - I(t)\, R_0\right) \tag{9}$$

where $h_{\text{init}}^{-1}(\cdot)$ is the inverse of $h_{\text{init}}(\cdot)$. Note that $\widehat{V}_s(t)$ is closely related but not identical to SoC, and (9) exactly holds only during an idling period. Thus, $h_{init}(\cdot)$ is a reasonable choice, but may not give a precise $\widehat{V}_s(t)$. However, the data-driven part in Section 4.2 will serve as a nonlinear correction to account for model errors of the linear part.

Since $G_1(q^{-1})$ is an integrator, it can be approximated by considering it as an SoC estimator and β_1 is determined by the inverse of the total capacity of a battery cell. Thus,

$$\text{SoC}(t) = G_1^1\left(q^{-1}\right) I(t) \tag{10}$$

Finally, β_2, β_3 can be obtained by a linear regression optimization on $G_2(q^{-1})$ as follows

$$\widehat{V}_s(t) - \text{SoC}(t) - V_s(0) = \frac{\beta_2 q^{-1}}{1 - \beta_3 q^{-1}} I(t) \tag{11}$$

The parameters for the discrete-time model can be transformed back to the physical parameters (Tian et al., 2021).

4.2 Estimation of the Nonlinear Part

We have obtained β_1, β_2, β_3, and R_0 of the VDC model from Section 4.1. The two input features $V_s(t)$ and SoC(t) can be then computed using (3) and (10). The rest is to identify the TT-cores $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d\}$ of the Volterra model in (7) by minimizing the following cost function

$$J = \frac{1}{2}\sum_{t=1}^{N}\left(y(t) - h\left(\mathbf{u}(t)\right)\right)^2 \tag{12}$$

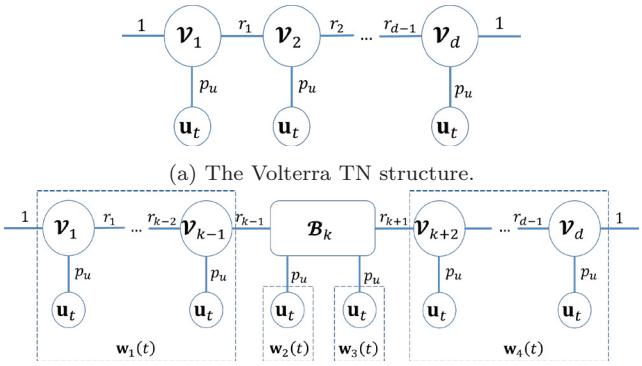
from the input feature vector

$$\mathbf{u}(t) = [V_s(t), \text{SoC}(t)]^{\mathrm{T}} \tag{13}$$

to the output electrode voltage $y(t)$ in (8)

$$y(t) = V(t) - I(t)R_0 \tag{14}$$

Following the techniques proposed in Stoudenmire and Schwab (2016); Batselier et al. (2017), the TT-cores can be estimated in an iterative fashion. In this paper, we are updating two neighboring TT-cores each time so that the TT-ranks can be adaptively adjusted to seek a low-rank representation, which is crucial to avoid overfitting. Denote $p_u = pM + 1$. The computation process in (7) is illustrated in Fig. 3a, which involves multiple 2-mode products. Suppose that we are updating the pair $\{\boldsymbol{\mathcal{V}}_k, \boldsymbol{\mathcal{V}}_{k+1}\}$ during an intermediate iteration. This pair as a whole can be treated as a 4-way bond core $\boldsymbol{\mathcal{B}}_k \in \mathbb{R}^{r_{k-1}\times p_u \times p_u \times r_{k+1}}$ as shown in Fig. 3b. The bond core $\boldsymbol{\mathcal{B}}_k$ is then updated whereas the remaining four parts $\{\mathbf{w}_1(t), \mathbf{w}_2(t), \mathbf{w}_3(t), \mathbf{w}_4(t)\}$ are fixed during this iteration.



(a) The Volterra TN structure.

(b) Estimate an intermediate bond core $\boldsymbol{\mathcal{B}}_k$ while fixing the remaining four parts $\{\mathbf{w}_1(t), \mathbf{w}_2(t), \mathbf{w}_3(t), \mathbf{w}_4(t)\}$.

Fig. 3. Illustration of a TN-based Volterra model.

More specifically, the voltage prediction $\hat{\mathbf{y}}(t)$ is

$$\hat{y}(t) = \boldsymbol{\mathcal{B}}_k \times_1 \mathbf{w}_1(t) \times_2 \mathbf{w}_2^{\mathrm{T}}(t) \times_3 \mathbf{w}_3^{\mathrm{T}}(t) \times_4 \mathbf{w}_4^{\mathrm{T}}(t) \tag{15}$$

where

$$\begin{aligned} \mathbf{w}_1(t) &= \left(\boldsymbol{\mathcal{V}}_1 \times_2 \mathbf{u}_t^{\mathrm{T}}\right) \cdots \left(\boldsymbol{\mathcal{V}}_{k-1} \times_2 \mathbf{u}_t^{\mathrm{T}}\right) \\ \mathbf{w}_2(t) &= \mathbf{w}_3(t) = \mathbf{u}_t \\ \mathbf{w}_4(t) &= \left(\boldsymbol{\mathcal{V}}_{k+2} \times_2 \mathbf{u}_t^{\mathrm{T}}\right) \cdots \left(\boldsymbol{\mathcal{V}}_d \times_2 \mathbf{u}_t^{\mathrm{T}}\right) \end{aligned} \tag{16}$$

The bond core $\boldsymbol{\mathcal{B}}_k$ can be estimated by solving a least-squares problem such that

$$\mathbf{Y} = \mathbf{W}\text{vec}\left(\boldsymbol{\mathcal{B}}_k\right) \tag{17}$$

where

$$\mathbf{W} = \begin{bmatrix} \mathbf{w}_4^{\mathrm{T}}(1) \otimes \mathbf{w}_3^{\mathrm{T}}(1) \otimes \mathbf{w}_2^{\mathrm{T}}(1) \otimes \mathbf{w}_1(1) \\ \mathbf{w}_4^{\mathrm{T}}(2) \otimes \mathbf{w}_3^{\mathrm{T}}(2) \otimes \mathbf{w}_2^{\mathrm{T}}(2) \otimes \mathbf{w}_1(2) \\ \vdots \\ \mathbf{w}_4^{\mathrm{T}}(N) \otimes \mathbf{w}_3^{\mathrm{T}}(N) \otimes \mathbf{w}_2^{\mathrm{T}}(N) \otimes \mathbf{w}_1(N) \end{bmatrix} \tag{18}$$

$$\mathbf{Y} = [\, y(1), y(2), \cdots, y(N) \,]^{\mathrm{T}}$$

and N is the number of available data points.

Each component in the pair $\{\boldsymbol{\mathcal{V}}_k, \boldsymbol{\mathcal{V}}_{k+1}\}$ is then updated by splitting up $\boldsymbol{\mathcal{B}}_k$ by using the singular value decomposition (SVD), where the TT-rank r_k is determined. We propose to use a ε-truncated SVD (Oseledets, 2011) where ε is the percentage of components to be omitted. For example, if the full SVD of a matrix

$$\mathbf{B} = \mathbf{L}\mathbf{S}\mathbf{Z}^{\mathrm{T}} = [\mathbf{L}_1\ \mathbf{L}_2] \begin{bmatrix} \mathbf{S}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{S}_2 \end{bmatrix} \begin{bmatrix} \mathbf{Z}_1^{\mathrm{T}} \\ \mathbf{Z}_2^{\mathrm{T}} \end{bmatrix} \tag{19}$$

then the ε-truncated SVD determines the rank r such that $\mathbf{B} \approx \mathbf{L}_1\mathbf{S}_1\mathbf{Z}_1^{\mathrm{T}}$ and the less significant components contained in $\mathbf{S}_2$ are omitted as many as possible so that their sum of squares is not greater than $\varepsilon^2||\mathbf{S}||_F^2$. Seeking a low-rank estimation of the TT-cores $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d\}$ can be regarded as introducing regularization during the optimization, which is common practice in machine learning

to overcome overfitting problems for complicated models with tons of parameters.

Algorithm 1: The alternating linear scheme with ε-truncated SVD

Input: Measurement data $\{\mathbf{u}(t), y(t)\}$ in (13) and (14) for $t = 1, \cdots, N$, memory length M, degree d, expected accuracy ε in percentage

Output: $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d\}$ in (7) minimizing (12)

1 Initialization: Construct $\mathbf{u}_t$ in (7); $r_0 \leftarrow 1, r_d \leftarrow 1$; initialize left orthogonal TN-cores $\{\boldsymbol{\mathcal{V}}_1, \cdots, \boldsymbol{\mathcal{V}}_d\}$ of ranks 1, i.e., $r_i \leftarrow 1, i = 1, 2, \cdots, d-1$; the starting index $k \leftarrow d-1$; sweeping direction R2L $\leftarrow$ 'left'

2 **while** *the termination criterion is not satisfied* **do**

  Compute $\{\mathbf{W}, \mathbf{Y}\}$ in (18)

  $\boldsymbol{\mathcal{B}}_k \leftarrow \text{reshape}(\text{pinv}(\mathbf{W})\mathbf{Y}, [r_{k-1}, p_u, p_u, r_{k+1}])$

  $\mathbf{B}_k \leftarrow \text{reshape}(\boldsymbol{\mathcal{B}}_k, [r_{k-1}p_u, p_u r_{k+1}])$

  $\{r_k, \mathbf{L}_1, \mathbf{S}_1, \mathbf{Z}_1\} \leftarrow \varepsilon$-truncated SVD on $\mathbf{B}_k$

  if *R2L is 'left'* **then**

    Split $\boldsymbol{\mathcal{B}}_k$ while keeping left orthogonal:

    $\boldsymbol{\mathcal{V}}_k \leftarrow \text{reshape}(\mathbf{L}_1\mathbf{S}_1, [r_{k-1}, p_u, r_k])$

    $\boldsymbol{\mathcal{V}}_{k+1} \leftarrow \text{reshape}(\mathbf{Z}_1^{\mathrm{T}}, [r_k, p_u, r_{k+1}])$

    if $k > 1$ **then**

      $k \leftarrow k - 1$

    else

      R2L$\leftarrow$'right'

  else

    Split $\boldsymbol{\mathcal{B}}_k$ while keeping right orthogonal:

    $\boldsymbol{\mathcal{V}}_k \leftarrow \text{reshape}(\mathbf{L}_1, [r_{k-1}, p_u, r_k])$

    $\boldsymbol{\mathcal{V}}_{k+1} \leftarrow \text{reshape}(\mathbf{S}_1\mathbf{Z}_1^{\mathrm{T}}, [r_k, p_u, r_{k+1}])$

    if $k < d-1$ **then**

      $k \leftarrow k + 1$

    else

      R2L$\leftarrow$'left'

The above discussions are summarized in Algorithm 1. The bond core $\boldsymbol{\mathcal{B}}_k$ is updated iteratively by sweeping the TN back and forth. The termination criterion for the iterative algorithm can be set as reaching either a minimum cost threshold or maximum iterations. The TT-cores are initialized to be either left or right orthogonal to facilitate the numerical stability of the sweeping algorithm (Holtz et al., 2012).

5. EXPERIMENTAL RESULTS

This section shows the experimental validation of the VDC model. The experiments were conducted on a Panasonic NCR18650B LiB cell using a PEC SBT4050 battery tester.

The estimate of the internal resistor is $R_0 = 0.08\ \Omega$. The total capacity in terms of the capacitance is estimated as 11,011 F. Thus, $\beta_1 = 1/11{,}011$ in $G_1(q^{-1})$. There are two training data sets. The first set, denoted as $\{I(t), V(t)\}$, is based on a variable-current profile (0∼6 A), which is shown in Fig. 5 along with the terminal voltage measurements. This data set conveys information about the dynamics of the model. The second data set is a formulated SoC-OCV step-wise mapping as shown in Fig. 6, which provides static information and is only used for the input of the Volterra model in the VDC model. The SoC-OCV data set is obtained as follows. First, 21 equidistantly data points $\{\text{SoC}_i, \text{OCV}_i\}, i = 1, \cdots, 21$ are sampled every 5% SoC

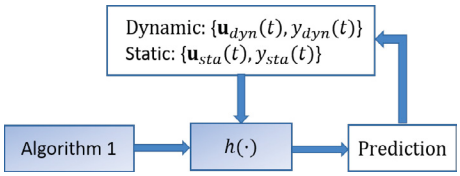
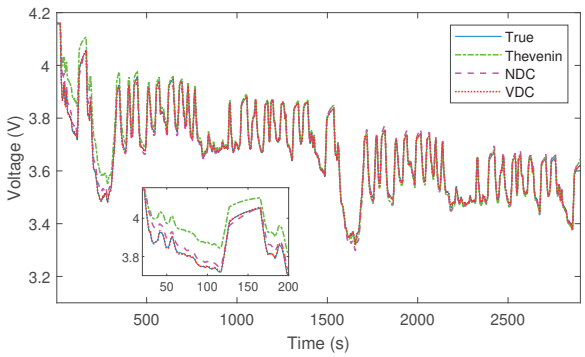
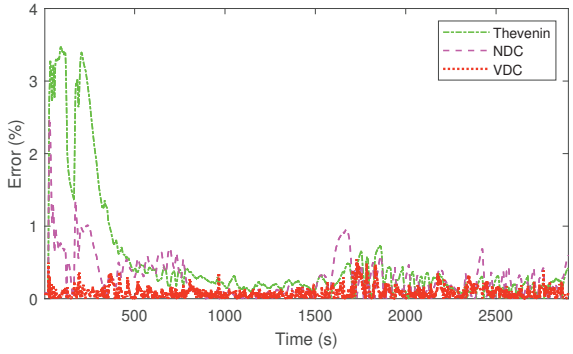


Fig. 4. Approximating the nonlinear function $h(\cdot)$ by using the TN-based Volterra model.



(a) The voltage prediction for the variable-current profile 0∼6 A.



(b) Fitting error in percentage.

Fig. 5. Illustration of the voltage prediction of the proposed model for the variable-current profile 0∼6 A.

step from the SoC-OCV curve. Then, each sampled point is augmented for 500 seconds (i.e., 500 points in our case). Following the discussion in Section 4.1, one can obtain that $\beta_2 = 7.9737 \times 10^{-4}, \beta_3 = 0.9757$ by using the first set of training data $\{I(t), V(t)\}$. By now, the estimation of the linear part has been completed. The next is to identify the TN-based Volterra model as discussed in Section 4.2. The input feature vector corresponding to the first set of the training data, denoted as $\mathbf{u}_{dyn}(t)$, can be now determined by (13). The output $y_{dyn}(t)$ is computed via (14). Here, the subscript "*dyn*" is shorthand for "dynamic". The input feature vector corresponding to the second set of the training data can be similarly determined and given by $\mathbf{u}_{sta}(t) = [\text{SoC}_i(t), \text{SoC}_i(t)]^{\mathrm{T}}$. The output $y_{sta} = \text{OCV}_i(t)$. The subscript "*sta*" is shorthand for "static". Note that $\mathbf{u}_{sta}(t)$ corresponds to the period when the battery is fully at rest and so $V_s(t) = \text{SoC}(t)$. Fig. 4 shows a schematic illustration of the above discussion.

The parameters of the TN-based Volterra model are set as follows: $M = 3$, $d = 5$, $\varepsilon = 0.01$. Thus, $p_u = pM + 1 = 7$. The estimated TT-ranks are $\{r_0 = 1, r_1 = 5, r_2 = 15, r_3 = 15, r_4 = 5, r_5 = 1\}$. Further, for comparison, we include the prediction results of the Thevenin (with

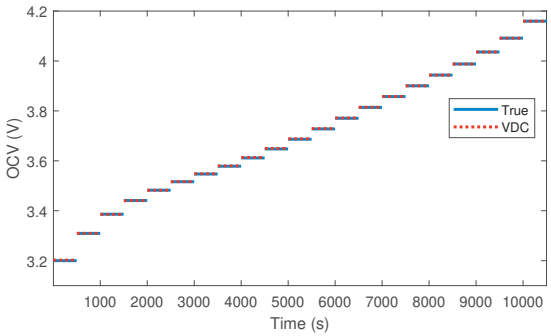
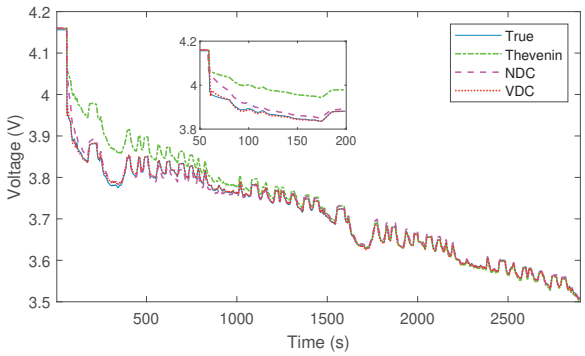


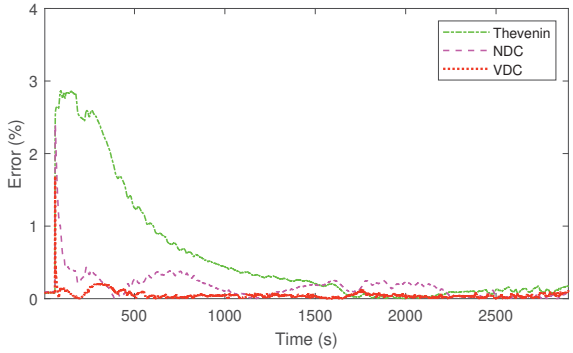
Fig. 6. The model prediction of SoC-OCV relationship every 5% SoC step from SoC = 0% to SoC = 100%.

2 RC circuits) and NDC (with 1 RC circuit) models, see Fig. 5 for the training results. The VDC model provides consistently superior prediction accuracy while the SoC is varying. The prediction of the NDC model is better than the Thevenin model but still less accurate than the VDC model. The performance of the VDC model results from its capability to capture more complex coupling dynamics and produce an inherently SoC-dependent prediction. The TT-ranks are adjusted adaptively during the iterations to seek a representation as low-rank as possible and avoid overfitting. The proposed VDC model is also precise in maintaining the SoC-OCV relationship as shown in Fig. 6. Thus, the VDC model is expected to have good performance in both dynamic and static predictions.

The identified models are further validated on a different data set based on a variable-current profile (1.5∼2.5 A) as shown in Fig. 7. The VDC model still shows the best prediction performance consistently. The low-rank representation obtained from the algorithm enables the identified model to be more robust to different current profiles and reduce the overfitting problems.



(a) The voltage prediction for the variable-current profile 1.5∼2.5 A.



(b) Fitting error in percentage.

Fig. 7. Illustration of the voltage prediction of the proposed model for the variable-current profile 1.5∼2.5 A.

6. CONCLUSION

High-performance models are essential to advanced BMS algorithms to guarantee the operating performance and safety of LiBs. While ECMs have proven their value and merits, they are often unable to meet the growing demands for predictive accuracy. To address this issue, this paper proposes to integrate an ECM with a TN-based Volterra model to create a hybrid model lying at the interface of physics- and data-based modeling. The new model, named as VDC, uses physical features from the ECM to inform the Volterra model such that the latter can accurately capture the nonlinear relationship in the data collected from LiBs. An iterative sweeping algorithm is derived to identify the TN-based Volterra model and can adaptively adjust the TT-ranks from the data to avoid overfitting. The experimental validation shows that the proposed model yields very high accuracy in voltage prediction. The model holds a promise for improving the BMS performance across various application domains.

REFERENCES

Ardeshiri, R.R., Balagopal, B., Alsabbagh, A., Ma, C., and Chow, M.Y. (2020). Machine learning approaches in battery management systems: State of the art: Remaining useful life and fault detection. In *2020 2nd IEEE International Conference on Industrial Electronics for Sustainable Energy Systems*, volume 1, 61–66. IEEE.

Azpicueta-Ruiz, L.A., Zeller, M., Figueiras-Vidal, A.R., Arenas-García, J., and Kellermann, W. (2011). Adaptive combination of Volterra kernels and its application to nonlinear acoustic echo cancellation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1), 97–110.

Batselier, K., Chen, Z., and Wong, N. (2017). Tensor network alternating linear scheme for MIMO Volterra system identification. *Automatica*, 84, 26–35.

Burt, P.M.S. and de Morais Goulart, J.H. (2018). Efficient computation of bilinear approximations and Volterra models of nonlinear systems. *IEEE Transactions on Signal Processing*, 66(3), 804–816.

Chemali, E., Kollmeyer, P.J., Preindl, M., and Emadi, A. (2018). State-of-charge estimation of li-ion batteries using deep neural networks: A machine learning approach. *Journal of Power Sources*, 400, 242–255.

Di Domenico, D., Stefanopoulou, A., and Fiengo, G. (2010). Lithium-ion battery state of charge and critical surface charge estimation using an electrochemical model-based extended Kalman filter. *Journal of Dynamic Systems, Measurement, and Control*, 132(6).

Farmann, A. and Sauer, D.U. (2016). A comprehensive review of on-board state-of-available-power prediction techniques for lithium-ion batteries in electric vehicles. *Journal of Power Sources*, 329, 123–137.

Forman, J.C., Moura, S.J., Stein, J.L., and Fathy, H.K. (2012). Genetic identification and fisher identifiability analysis of the Doyle-Fuller-Newman model from experimental cycling of a LiFePO4 cell. *Journal of Power Sources*, 210, 263–275.

Goodenough, J.B. and Park, K.S. (2013). The li-ion rechargeable battery: a perspective. *Journal of the American Chemical Society*, 135(4), 1167–1176.

Hannan, M.A., Lipu, M.H., Hussain, A., Ker, P.J., Mahlia, T., Mansor, M., Ayob, A., Saad, M.H., and Dong, Z. (2020). Toward enhanced state of charge estimation of lithium-ion batteries using optimized machine learning techniques. *Scientific Reports*, 10(1), 1–15.

Hatzell, K.B., Sharma, A., and Fathy, H.K. (2012). A survey of long-term health modeling, estimation, and control of lithium-ion batteries: Challenges and opportunities. In *2012 American Control Conference*, 584–591. IEEE.

Holtz, S., Rohwedder, T., and Schneider, R. (2012). The alternating linear scheme for tensor optimization in the tensor train format. *SIAM Journal on Scientific Computing*, 34(2), A683–A713.

Hu, X., Li, S.E., and Yang, Y. (2015). Advanced machine learning approach for lithium-ion battery state estimation in electric vehicles. *IEEE Transactions on Transportation electrification*, 2(2), 140–149.

Hu, Y. and de Callafon, R.A. (2017). Optimal weighting for covariance based realization algorithm. In *2017 IEEE 56th Annual Conference on Decision and Control*, 5274–5279. IEEE.

Hu, Y., Jiang, Y., and de Callafon, R.A. (2020). Variance reduction in covariance based realization algorithm with application to closed-loop data. *Automatica*, 113, 108683.

Hu, Y., Konakalla, S.A.R., and de Callafon, R.A. (2018). Covariance based estimation for reduced order models of microgrid power flow dynamics. In *2018 IFAC 18th Symposium on System Identification*, 903–908. IFAC.

Hu, Y., Tan, L., and de Callafon, R.A. (2019). Persistent excitation condition for MIMO Volterra system identification with gaussian distributed input signals. In *2019 IEEE 58th Annual Conference on Decision and Control*, 1752–1757. IEEE.

Johnson, V.H., Pesaran, A.A., and Sack, T. (2001). Temperature-dependent battery models for high-power lithium-ion batteries. Technical report, National Renewable Energy Lab., Golden, CO (US).

Li, J., Adewuyi, K., Lotfi, N., Landers, R.G., and Park, J. (2018). A single particle model with chemical/mechanical degradation physics for lithium ion battery state of health (SOH) estimation. *Applied Energy*, 212, 1178–1190.

Lin, X., Perez, H.E., Mohan, S., Siegel, J.B., Stefanopoulou, A.G., Ding, Y., and Castanier, M.P. (2014). A lumped-parameter electro-thermal model for cylindrical batteries. *Journal of Power Sources*, 257, 1–11.

Liu, J., Bao, Z., Cui, Y., Dufek, E.J., Goodenough, J.B., Khalifah, P., Li, Q., Liaw, B.Y., Liu, P., Manthiram, A., Meng, Y.S., Subramanian, V.R., Toney, M.F., Viswanathan, V.V., Whittingham, M.S., Xiao, J., Xu, W., Yang, J., Yang, X.Q., and Zhang, J.G. (2019). Pathways for practical high-energy long-cycling lithium metal batteries. *Nature Energy*, 4(3), 180–186.

Ljung, L. (1999). *System Identification: Theory for the User*. Prentice-Hall, second edition.

Moura, S.J., Argomedo, F.B., Klein, R., Mirtabatabaei, A., and Krstic, M. (2017). Battery state estimation for a single particle model with electrolyte dynamics. *IEEE Transactions on Control Systems Technology*, 25(2), 453–468.

Murphey, Y.L., Park, J., Chen, Z., Kuang, M.L., Masrur, M.A., and Phillips, A.M. (2012). Intelligent hybrid vehicle power control—part i: Machine learning of optimal vehicle power. *IEEE Transactions on Vehicular Technology*, 61(8), 3519–3530.

Murphy, K.P. (2012). *Machine learning: a probabilistic perspective*. The MIT Press, Cambridge, MA.

Oseledets, I.V. (2011). Tensor-train decomposition. *SIAM Journal on Scientific Computing*, 33(5), 2295–2317.

Pinson, M.B. and Bazant, M.Z. (2012). Theory of sei formation in rechargeable batteries: capacity fade, accelerated aging and lifetime prediction. *Journal of the Electrochemical Society*, 160(2), A243–A250.

Plett, G.L. (2015). *Battery Management Systems, Volume II: Equivalent-Circuit Methods*. Artech House.

Rahn, C.D. and Wang, C.Y. (2013). *Battery Systems Engineering*. Wiley.

Ramadesigan, V., Northrop, P.W.C., De, S., Santhanagopalan, S., Braatz, R.D., and Subramanian, V.R. (2012). Modeling and simulation of lithium-ion batteries from a systems engineering perspective. *Journal of The Electrochemical Society*, 159(3), R31–R45.

Santhanagopalan, S., Guo, Q., Ramadass, P., and White, R.E. (2006). Review of models for predicting the cycling performance of lithium ion batteries. *Journal of Power Sources*, 156(2), 620–628.

Scrosati, B., Garche, J., and Tillmetz, W. (2015). *Advances in Battery Technologies for Electric Vehicles*. Woodhead Publishing, Waltham, MA.

Stoudenmire, E. and Schwab, D.J. (2016). Supervised learning with tensor networks. In *Advances in Neural Information Processing Systems*, 4799–4807.

Tian, N., Fang, H., Chen, J., and Wang, Y. (2021). Nonlinear double-capacitor model for rechargeable batteries: Modeling, identification, and validation. *IEEE Transactions on Control Systems Technology*, 29(1), 370–384.

Verhaegen, M. and Verdult, V. (2007). *Filtering and System Identification: A Least Squares Approach*. Cambridge University Press, New York, NY.

Wambacq, P. and Sansen, W.M. (1998). *Distortion Analysis of Analog Integrated Circuits*. Springer Science & Business Media, New York, NY.

Wang, Y., Fang, H., Zhou, L., and Wada, T. (2017). Revisiting the state-of-charge estimation for lithium-ion batteries: A methodical investigation of the extended Kalman filter approach. *IEEE Control Systems Magazine*, 37(4), 73–96.

Xia, B., Zhao, X., De Callafon, R., Garnier, H., Nguyen, T., and Mi, C. (2016). Accurate lithium-ion battery parameter estimation with continuous-time system identification methods. *Applied Energy*, 179, 426–436.

Yang, D., Zhang, X., Pan, R., Wang, Y., and Chen, Z. (2018). A novel Gaussian process regression model for state-of-health estimation of lithium-ion battery using charging curve. *Journal of Power Sources*, 384, 387–395.

Zou, C., Manzie, C., and Nešić, D. (2018). Model predictive control for lithium-ion battery optimal charging. *IEEE/ASME Transactions on Mechatronics*, 23(2), 947–957.